

# **Problemas de la $p$ -mediana Aplicación**



**Sofía Led Jiménez**  
Trabajo de Fin de Grado en Matemáticas  
Universidad de Zaragoza

Directora del trabajo: Herminia I. Calvete Fernández  
Julio de 2018



# Prólogo

El problema de la  $p$ -mediana es fundamental para gran parte de la teoría de localización discreta. Está motivado por una serie de situaciones reales, como por ejemplo tener que instalar varias plantas en algunos puntos de un sistema de transporte para minimizar los costos de producción y envío o tener que instalar diferentes hospitales en una ciudad para que la mayor parte de su población quede cubierta en caso de necesitar uno. En definitiva, el problema de la  $p$ -mediana trata de localizar  $p$  instalaciones dentro de una red de manera que se minimice la distancia ponderada entre los puntos de demanda y la instalación abierta más cercana.

En el primer capítulo de esta memoria, se estudian los problemas de optimización lineal y lineal entera, y se da una idea de varios algoritmos para la resolución de los mismos.

En el segundo capítulo, se estudia el problema de la  $p$ -mediana. Se analiza en primer lugar el caso más simple, el problema de la mediana, presentando los resultados teóricos más importantes. A continuación, se estudia el problema de la  $p$ -mediana y se formula como un problema de optimización lineal entera. Finalmente, se presentan varios procedimientos para resolver el problema y se formulan varias extensiones del mismo.

En el tercer capítulo se aplica el problema de la  $p$ -mediana al caso particular de la posible localización de ciertas instalaciones en la Comunidad Autónoma de Aragón y se presenta un estudio reciente realizado en Pekín sobre la localización de estaciones de carga para vehículos eléctricos.



# Summary

This work addresses one of the basic models in discrete location theory: the  $p$ -median problem. The problem is motivated by a number of potential applications. One example is to locate several plants at some points of a transportation system to minimize production and shipment costs.

Let us consider a finite network  $N$  formed by a set of vertices  $V = \{v_1, \dots, v_n\}$  and by a set of edges  $E$ , which have attached nonnegative weights. The edge  $b_i$  has associated a weight  $h_i$  and the vertex  $v_i$  has associated a weight  $w_i$ . It is defined the distance  $d(x, y)$  between two points  $x, y$  in  $N$ , as the length of the sortest path in  $N$  between points  $x$  and  $y$ , where the length of a path is the sum of the weight of the edges of that path.

Firstly, we consider the median problem, which consists in selecting one point of  $N$  in order to optimize a function which is distance-dependent with respect to given points of the network, that is, a point  $x_0 \in N$  is a median if and only if

$$\sum_{i=1}^n w_i d(v_i, x_0) \leq \sum_{i=1}^n w_i d(v_i, x) \quad \forall x \in N$$

Hakimi [10] proved a theorem in which he showed that an optimal solution can always be found in the set of vertices.

**Theorem 0.1.** *An absolute median of a graph is always at a vertex of a graph.*

Due to this fact, a discrete representation of the continuous problem can be provided.

So far, we have considered the location of a single facility. The next step is to locate  $p$  facilities on the network, where  $p$  is an integer larger than one. The objective of the  $p$ -median problem is to locate  $p$  facilities so that the sum of the weighted distances between the demand vertices and the selected facilities is minimized. In 1965, Hakimi [11] showed that the set of possible sites for the facilities can be restricted to the set of vertices. Given this property, the  $p$ -median problem can be expressed as a mixed-integer linear program as follows:

$$\min_{Y, X} F(Y, X) = \sum_{i=1}^n \sum_{j=1}^n w_i d_{ij} x_{ij} \quad (1a)$$

subject to

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (1b)$$

$$\sum_{j=1}^n y_j = p \quad (1c)$$

$$0 \leq x_{ij} \leq y_j, \quad i = 1, \dots, n, \quad j = 1, \dots, n \quad (1d)$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (1e)$$

where  $d_{ij}$  is the distance between vertices  $v_i$  and  $v_j$ , the variable  $y_j$  is set equal to one if  $v_j$  belongs to the  $p$ -median and to zero otherwise and the variable  $x_{ij}$  is the fraction of the demand arising at  $v_i$  satisfied by the facility at  $v_j$  (if any).  $Y$  denotes the vector of the variables  $y_j$  and  $X$  is the matrix of the variables  $x_{ij}$ . The constraints (1b) mean that the demand at  $v_i$  must be met by some facility. The constraint (1c)

ensures that exactly  $p$  vertices belong to the  $p$ -median. The third group (1d) prevents an assignment to a vertex not in the  $p$ -median, and ensures the nonnegativity of the variables  $x_{ij}$ . Constraints (1e) restrict each variable  $y_j$  to be a binary variable.

Although the problem is classified as NP-hard, there are different heuristic algorithms that in practice solve the  $p$ -median problem efficiently. They can be divided into two groups:

1. Classical heuristics:

- Greedy method.
- Stingy method.
- Neighborhood Search method.
- Interchange method.
- Lagrangian Relaxation method.

2. Metaheuristics:

- Tabu Search method.
- Scatter Search method.
- Hybrid method.

In some cases, it can be interesting to add some constraints to the problem or to modify the objective function to model other situations. Different variants of the  $p$ -median problem arise, of which we study some of them.

The capacitated  $p$ -median problem has the same objective than the  $p$ -median problem, but in this case, there are constraints that force the facilities to have a maximum capacity. The problem can be formulated as a mixed-integer linear program as follows:

$$\begin{aligned} \min_{Y,X} F(Y,X) &= \sum_{i=1}^n \sum_{j=1}^n d_{ij} z_{ij} \\ \text{subject to} \\ \sum_{j=1}^n z_{ij} &= w_i, \quad i = 1, \dots, n \\ \sum_{j=1}^n y_j &= p \\ z_{ij} &\geq 0, \quad i = 1, \dots, n, j = 1, \dots, n \\ z_{ij} &\leq w_i y_j, \quad i = 1, \dots, n, j = 1, \dots, n \\ \sum_{i=1}^n z_{ij} &\leq C, \quad j = 1, \dots, n \\ y_j &\in \{0, 1\}, \quad j = 1, \dots, n \end{aligned}$$

where the decision variable  $z_{ij}$  describes the amount of demand at  $v_i$  that is serviced by a facility placed at  $v_j$  and  $C$  is the maximum capacity of demand that can serve a facility.

Other variant of the  $p$ -median problem is the fixed cost facility location problem, whose objective is to minimize the total costs and to find the number of facilities and their locations. Formally, the problem can be expressed as a mixed-integer linear program:

$$\min_{Y,X} F(Y,X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij} + \sum_{j=1}^n f_j y_j$$

subject to

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= 1, \quad i = 1, \dots, n \\ 0 \leq x_{ij} &\leq y_j, \quad i = 1, \dots, n, \quad j = 1, \dots, n \\ y_j &\in \{0, 1\}, \quad j = 1, \dots, n \end{aligned}$$

where  $c_{ij}$  is the cost of supplying customers at  $v_i$  from facility located in  $v_j$  and  $f_j$  is the fixed set-up costs for locating the facility in  $v_j$ .

Lastly, the maximal covering problem is another variant which consists on maximize the number of customers served. The problem can be formulated as follows:

$$\max_{Y,X} F(Y,X) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}x_{ij}$$

subject to

$$\begin{aligned} \sum_{j=1}^n x_{ij} &\leq 1, \quad i = 1, \dots, n \\ \sum_{j=1}^n y_j &= p \\ 0 \leq x_{ij} &\leq y_j, \quad i = 1, \dots, n, \quad j = 1, \dots, n \\ y_j &\in \{0, 1\}, \quad j = 1, \dots, n \end{aligned}$$

where the parameter  $w_{ij}$  is defined as:

$$w_{ij} = \begin{cases} w_i & \text{si } d_{ij} \leq \bar{d}_i \\ 0 & \text{si } d_{ij} > \bar{d}_i \end{cases}$$

and  $\bar{d}_i$  is the maximum distance at which a facility can be located of the vertex  $v_i$ .

The work also addresses the  $p$ -median problem on a tree, which can be solved in polynomial time.

Finally, two applications of the  $p$ -median problem are provided. One of them is an application to the possible location of certain facilities in Aragón. The customers set and the possible location set are the ‘comarcas’ (administrative district) of Aragón, represented by their administrative capitals. We use an extension of Excel called ‘VRP Solver’ to get the distance between each capital and we use a program called *CPLEX* to solve the  $p$ -median model that we have constructed. We make several executions of the model changing the value of  $p$  to evaluate and to compare the results. For instance, for  $p = 3$ , the facilities are located in Zaragoza, Huesca and Teruel, the three province capitals, and for  $p = 6$  the facilities are located in Alcañiz, Huesca, La Almunia de Doña Godina, Monzón, Teruel and Zaragoza.

The other application is a study carried out by He et al. [12] in Beijing on the location of charging stations for electric vehicles. That study compares the optimum results of three models: the set covering model, the maximal covering location model, and the  $p$ -median model. The study conclude that the results of the  $p$ -median problem are better to theirs interests, which are to promote the electric vehicles consumption to improve the city’s air quality.





# Índice general

<b>Prólogo</b>	<b>III</b>
<b>Summary</b>	<b>V</b>
<b>1. Optimización</b>	<b>1</b>
1.1. Optimización lineal . . . . .	1
1.1.1. Condiciones de optimalidad . . . . .	2
1.1.2. Resolución del problema de optimización lineal: Método simplex . . . . .	3
1.2. Optimización lineal entera . . . . .	3
1.2.1. Algoritmo de ramificación y acotación . . . . .	4
1.2.2. Método del plano de corte . . . . .	4
<b>2. Problemas de la <math>p</math>-mediana</b>	<b>5</b>
2.1. Introducción . . . . .	5
2.2. Historia y desarrollo . . . . .	5
2.3. El problema de la mediana . . . . .	5
2.4. El problema de la $p$ -mediana como problema de optimización entera . . . . .	8
2.5. Algoritmos . . . . .	8
2.5.1. Método Greedy . . . . .	9
2.5.2. Método del Vecindario o Alterno . . . . .	9
2.5.3. Método Stingy . . . . .	10
2.5.4. Método de intercambio . . . . .	10
2.5.5. Método de Lagrange . . . . .	11
2.5.6. Método de búsqueda tabú ( $TS$ ) . . . . .	12
2.5.7. Método de búsqueda dispersa ( $SS$ ) . . . . .	12
2.5.8. Método híbrido . . . . .	13
2.6. Variantes del problema de la $p$ -mediana . . . . .	13
2.6.1. El problema de la $p$ -mediana en forma de árbol . . . . .	13
2.6.2. Planificación de instalaciones con capacidad . . . . .	14
2.6.3. El problema de localización de instalaciones con costes fijos . . . . .	14
2.6.4. Problema de cobertura máxima . . . . .	15
<b>3. Aplicaciones</b>	<b>17</b>
3.1. Introducción . . . . .	17
3.2. Una aplicación en la Comunidad Autónoma de Aragón . . . . .	17
3.2.1. Recopilación de datos . . . . .	17
3.2.2. Creación del modelo . . . . .	18
3.2.3. Evaluación de los resultados . . . . .	20
3.3. Una aplicación a la localización de estaciones de recarga . . . . .	22
3.3.1. Demanda y localizaciones . . . . .	23
3.3.2. Resultados . . . . .	23

<b>Bibliografía</b>	<b>25</b>
<b>Anexo</b>	<b>27</b>

# Capítulo 1

## Optimización

Los problemas de optimización son aquellos que se ocupan de elegir la decisión óptima de un problema, es decir, tratan de encontrar cual es el máximo o mínimo de una determinada función sujeto a unas condiciones que nos da el problema.

### 1.1. Optimización lineal

Un problema de optimización lineal es un problema de optimización en el cual se trata de maximizar o minimizar una función lineal de las variables decisión y donde cada restricción puede escribirse como una ecuación o inecuación lineal. Las variables pueden estar restringidas a ser no negativas, no positivas o no estar restringidas en signo.

Se presenta a continuación el problema de optimización lineal de máximo en forma estándar, el cual consiste en determinar los valores de las variables decisión  $x_1, \dots, x_n$  tales que:

$$\begin{aligned} \max \quad Z &= c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{sujeto a} \quad & \\ & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ & \dots \\ & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \\ & x_i \geq 0, \quad i = 1, \dots, n. \end{aligned} \tag{1.1}$$

Escrito en forma matricial el problema en forma estándar queda:

$$\begin{aligned} \max \quad & cX \\ \text{sujeto a} \quad & AX = b \\ & X \geq 0 \end{aligned} \tag{1.2}$$

donde:

- $X^t = (x_1, \dots, x_n)^t$  es el vector traspuesto de variables de dimensión  $1 \times n$ .
- $b^t = (b_1, \dots, b_m)^t$  es el vector traspuesto de recursos de dimensión  $1 \times m$ .
- $c = (c_1, \dots, c_n)$  es el vector de coeficientes de la función objetivo de dimensión  $1 \times n$ .
- $A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$  es la matriz de coeficientes tecnológicos de dimensión  $m \times n$ .

Suponemos que  $m < n$  y que  $\text{rango}(A) = m$ .

- $Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$  es la función objetivo.

**Nota 1.1.** En el caso de que el problema sea de mínimo, lo único que cambia es que se escribe min en vez de max en la función objetivo.

### 1.1.1. Condiciones de optimalidad

En esta sección se enuncia un teorema que da una condición necesaria y suficiente para que el problema de optimización lineal tenga solución. Previamente, se introducen algunos conceptos clave para su comprensión.

#### Definiciones.

- Región factible o región de factibilidad: Poliedro  $S$  de  $\mathbb{R}^n$  definido por las restricciones del problema.

$$S = \{X \in \mathbb{R}^n : AX = b, X \geq 0\}$$

- Un punto  $X \in S$  es un punto extremo de  $S$  si no puede ponerse como una combinación convexa de puntos distintos de  $S$ .
- Un vector  $d$  de  $\mathbb{R}^n$  no nulo es una dirección de  $S$  si  $X + \lambda d \in S$ ,  $\lambda \geq 0$  y  $X \in S$ .
- Dos direcciones  $d_1$  y  $d_2$  son distintas si no son proporcionales.
- Una dirección  $d$  de  $S$  es extrema si no puede escribirse como una combinación lineal positiva de dos direcciones distintas.
- Una solución básica es una solución con a lo sumo  $m$  valores no nulos tal que los vectores columna de la matriz  $A$  asociados a las componentes no nulas son linealmente independientes.
- Una solución factible básica (s.f.b.) es una solución simultáneamente factible y básica. Las s.f.b. son los puntos extremos de la región de factibilidad.
- Dos s.f.b. son adyacentes si sus conjuntos de variables básicas tienen  $m - 1$  variables básicas en común.

Se enuncia a continuación el teorema mencionado anteriormente.

**Teorema 1.2.** Sea el problema de optimización lineal de máximo en forma estándar (1.2), siendo  $X: n \times 1$ ,  $A: m \times n$  de rango  $m$ ,  $b: m \times 1$  y  $c: 1 \times n$ .

Supondremos que la región de factibilidad es no vacía.

Sean  $X_1, X_2, \dots, X_k$  los puntos extremos y  $d_1, d_2, \dots, d_h$  las direcciones extremas de la región factible.

Una condición necesaria y suficiente para que el problema tenga solución óptima es que

$$cd_i \leq 0, \quad i = 1, \dots, h.$$

Además, si el problema tiene solución óptima, existe un punto extremo que es solución óptima del problema.

**Nota 1.3.** Para el caso en el que el problema sea de mínimo la condición necesaria y suficiente es:

$$cd_i \geq 0, \quad i = 1, \dots, h.$$

Hay que destacar que el conjunto de puntos extremos del poliedro  $S$  es finito, así como que siempre existe al menos un punto extremo. Además, existe una dirección extrema si y solo si el poliedro es no acotado. Se puede encontrar la demostración de este teorema, así como más información sobre el problema de optimización lineal en [1, 2].

### 1.1.2. Resolución del problema de optimización lineal: Método simplex

Si el problema de optimización lineal tiene solución óptima finita, teniendo en cuenta el teorema 1.2 que proporciona las condiciones de optimalidad, para encontrar una solución óptima basta concentrarse en las s.f.b. (puntos extremos). El algoritmo simplex se basa en este hecho y proporciona un método sistemático para resolver un problema de optimización lineal.

Partiendo de una s.f.b., el algoritmo itera cambiando de una s.f.b. a otra s.f.b. adyacente con valor mejor (o al menos no peor) de la función objetivo. El proceso continúa hasta que se alcanza una s.f.b. óptima o se encuentra una dirección extrema  $d$  para la que no se verifica la condición del teorema 1.2, si el problema es de máximo, o la de la nota 1.3, si el problema es de mínimo. Si se encuentra esta dirección, el algoritmo concluye indicando que el problema es no acotado.

Cabe destacar que para poder utilizar el método simplex, el problema tiene que estar formulado en forma estándar. Otros métodos de solución del problema pueden encontrarse en [1].

## 1.2. Optimización lineal entera

Un problema de optimización lineal entera es un problema de optimización en el que algunas de las variables están restringidas a tomar valores enteros, y tal que la función objetivo y las restricciones son funciones lineales de las variables. Estos problemas se clasifican de acuerdo con el tipo y el número de variables enteras. Con la notación introducida, el problema se dice:

- **Entero puro (PILP)** si todas las variables del problema están restringidas a ser enteras.

$$\begin{aligned} \max \quad & cX \\ \text{sujeto a} \quad & AX \leq b \\ & X \geq 0, X \in \mathbb{Z}^n \end{aligned}$$

La región factible en este caso es  $S_E = \{X \in \mathbb{Z}_+^n : AX \leq b\}$ .

- **Entero mixto (MILP)** si en el problema coexisten variables enteras y reales.

$$\begin{aligned} \max \quad & cX + hY \\ \text{sujeto a} \quad & AX + GY \leq b \\ & X \geq 0, X \in \mathbb{Z}^n \\ & Y \geq 0, Y \in \mathbb{R}^p \end{aligned} \tag{1.3}$$

donde  $c = (c_1, \dots, c_n)$ ,  $h = (h_1, \dots, h_p)$ ,  $A = (a_{ij})$  matriz  $m \times n$  y  $G = (g_{ij})$  matriz  $m \times p$ .

En este caso la región factible es  $S_E = \{(X, Y) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : AX + GY \leq b\}$ .

- **Entero binario** si todas las variables del problema están restringidas a tomar los valores 0 ó 1. En este caso, la región factible es  $S_E = \{X \in \{0, 1\}^n : AX \leq b\}$ .

A continuación, se introducen dos algoritmos para resolver problemas de optimización entera. Un planteamiento más completo del problema puede encontrarse en [5].

La idea general de los métodos de resolución de problemas de optimización entera consiste en resolver problemas lineales que se obtienen relajando la restricción de integridad.

Dado el programa entero mixto (1.3), sea  $(X^*, Y^*)$  una solución óptima y  $Z^*$  el valor óptimo del MILP. Sea  $P_0 := \{(X, Y) \in \mathbb{R}_+^n \times \mathbb{R}_+^p : AX + GY \leq b\}$  la relajación natural de  $S_E$  y sea  $(X^0, Y^0)$  una solución óptima básica del problema sin la restricción de integridad. Si  $(X^0, Y^0) \in S_E$ , entonces esta es la solución óptima del problema MILP. En caso contrario, el proceso continúa resolviendo problemas lineales sobre recintos modificados, como se explica a continuación.

### 1.2.1. Algoritmo de ramificación y acotación

En primer lugar, se resuelve el problema relajado planteado anteriormente. Si  $(X^0, Y^0) \notin S_E$ , se aplica el algoritmo de ramificación y acotación.

- **Ramificación:** se divide la región de factibilidad original  $P_0$  en dos subconjuntos, eliminando una parte de  $P_0$  que no incluye soluciones factibles que verifiquen las restricciones de integridad.

- Primero se selecciona para ramificar una variable básica que debiendo ser entera no lo es actualmente. Sea esta variable  $x_{B_r}$ , que toma actualmente el valor  $\bar{b}_r$ .
- Se ramifica el problema construyendo a partir de él dos subproblemas que incorporan, respectivamente, las restricciones:

$$P1 : \quad x_{B_r} \leq [\bar{b}_r] \qquad P2 : \quad x_{B_r} \geq [\bar{b}_r] + 1$$

donde  $[\cdot]$  es la parte entera.

Se eliminan del estudio los valores  $[\bar{b}_r] < x_{B_r} < [\bar{b}_r] + 1$ , ninguno de los cuales es entero.

- Se resuelven estos dos nuevos problemas de optimización lineal y se aplica la acotación.
- **Acotación:** el valor óptimo de la función objetivo de cada subproblema creado es una cota superior, si el problema es de máximo, o una cota inferior, si el problema es de mínimo, del valor óptimo de la función objetivo de las soluciones del problema entero que pertenecen a su región de factibilidad. Estas cotas permiten descartar del estudio algunos subproblemas que no contienen la solución óptima entera y localizar la solución entera.

Un subproblema deja de ramificarse cuando se cumple alguna de las condiciones siguientes:

- Su solución óptima es entera.
- No contiene soluciones enteras con mejor valor de la función objetivo que alguna solución entera ya obtenida.
- Es no factible.

El algoritmo se itera hasta que no se pueden ramificar más problemas.

En [5, 22] pueden estudiarse otras formas de ramificar, así como distintos procedimientos para seleccionar la variable de ramificación.

### 1.2.2. Método del plano de corte

Se comienza resolviendo el problema entero relajado. Si se tiene que  $(X^0, Y^0) \notin S_E$ , se aplica el método del plano de corte.

La idea del método es encontrar una desigualdad  $\alpha X + \gamma Y \leq \beta$  que se satisfaga  $\forall (X, Y) \in S_E$  y tal que  $\alpha X^0 + \gamma Y^0 > \beta$ ; es decir,  $\alpha X + \gamma Y \leq \beta$  es un plano de corte que separa  $(X^0, Y^0)$  de  $S_E$ .

Sea  $\alpha X + \gamma Y \leq \beta$  un plano de corte y definimos

$$P_1 := P_0 \cap \{(X, Y) : \alpha X + \gamma Y \leq \beta\}$$

Dado que  $S_E \subseteq P_1 \subset P_0$ , la relajación del problema MILP basada en  $P_1$  es más fuerte que la relajación natural  $P_0$ , en el sentido de que el valor óptimo del problema lineal  $\max\{cX + hY : (X, Y) \in P_1\}$  es una cota superior de  $Z^*$  y es una cota inferior de  $Z^0$ , porque la solución óptima  $(X^0, Y^0)$  de la relajación  $P_0$  no pertenece a  $P_1$ .

La aplicación recursiva de esta idea conduce al enfoque del plano de corte, cuyas etapas son:

Empezando por  $i = 0$ , repetir:

**Paso recursivo:** resolver el programa lineal  $\max\{cX + hY : (X, Y) \in P_i\}$ .

- Si la solución básica óptima asociada a  $P_i$ ,  $(X^i, Y^i)$ , pertenece a  $S_E$ , entonces parar.

- En otro caso, encontrar un plano de corte  $\alpha X + \gamma Y \leq \beta$  que separe  $(X^i, Y^i)$  de  $S_E$ .

Definir el conjunto  $P_{i+1} := P_i \cap \{(X, Y) : \alpha X + \gamma Y \leq \beta\}$  y repetir el paso recursivo.

En [5, 22] pueden estudiarse distintas formas de generar planos de corte.

## Capítulo 2

# Problemas de la $p$ -mediana

### 2.1. Introducción

El problema de la  $p$ -mediana es uno de los modelos básicos en la teoría de localización discreta y se remonta al trascendental trabajo de Hakimi en 1964-1965. El objetivo del problema es localizar  $p$  instalaciones de manera que se minimice la suma de las distancias ponderadas o costes de transporte entre los vértices con demanda y las instalaciones seleccionadas. Al igual que la mayoría de los problemas de localización, está clasificado como NP-hard y, por ello, para resolverlo se suelen usar métodos heurísticos, ya que son capaces de dar rápidamente una aproximación a la solución. Por otro lado, se han desarrollado algoritmos que resuelven el problema en un tiempo polinomial cuando está planteado en forma de árbol. El problema de la  $p$ -mediana y sus extensiones son útiles para modelar muchas situaciones reales, como la localización de plantas industriales, almacenes e instalaciones públicas.

### 2.2. Historia y desarrollo

A lo largo de la historia se han planteado diferentes problemas que acabaron dando lugar al problema de la  $p$ -mediana tal y como se conoce ahora.

Ya en el siglo XVII, Fermat planteó el problema en el cual, dado un triángulo, se buscaba un punto en el plano tal que la suma de las distancias euclídeas desde cada vértice hasta el punto fuera mínima.

El primer problema de localización surgió a principios del siglo XX. Alfred Weber añadió al problema formulado anteriormente por Fermat ponderaciones en cada uno de los tres vértices para simular así la demanda de los clientes. En este caso, la mejor ubicación para que la instalación satisficiera las demandas se correspondió con el baricentro.

A principios de la década de 1960, Hakimi [10, 11], desarrolló problemas similares para encontrar medianas en una red o un grafo. Denominó ‘mediana absoluta’ al punto del grafo que minimiza la suma de las distancias ponderadas entre ese punto y los vértices del grafo, y probó que una mediana absoluta óptima siempre se localiza en un vértice del grafo, proporcionando así una representación discreta de un problema continuo. Además, generalizó también el problema de la mediana absoluta para hallar  $p$  medianas en un grafo que minimizaran la suma de las distancias ponderadas y demostró que siempre hay una colección de  $p$  vértices que minimizan la función objetivo. De esta forma, volvió a proporcionar una representación discreta de un problema continuo al restringir la búsqueda a los vértices.

Desde su trabajo, el problema de la  $p$ -mediana ha ido de la mano de la teoría de localización, convirtiéndose en uno de los modelos de localización de instalaciones más comunes [16, 17].

### 2.3. El problema de la mediana

Para poder enunciar el problema de la mediana, previamente se tienen que introducir una serie de conceptos.

Se considera un grafo (red) finito no dirigido  $N$  formado por un conjunto de vértices  $V = \{v_1, \dots, v_n\}$  y un conjunto de ejes  $E$ , los cuales tienen asociados unos pesos no negativos. El peso  $h_i$  asociado al eje  $b_i$  de  $N$  representa la longitud del eje; el peso  $w_i$  asociado al vértice  $v_i$  de  $N$  representa la demanda que hay en ese vértice. Un punto  $x \in N$  es un punto a lo largo de un eje de  $N$ , y puede ser o no ser un vértice del grafo. La distancia de cualesquiera dos puntos  $x$  e  $y$  de  $N$  se representa por  $d(x, y)$  y se define como la longitud del camino más corto en  $N$  entre  $x$  y  $y$ , donde la longitud del camino es la suma de los pesos de los ejes que lo forman.

La matriz de distancia  $D = [d_{ij}]$  en el grafo  $N$  se define como:

$$d_{ij} = \begin{cases} d(v_i, v_j) & \text{para } i, j = 1, 2, \dots, n, \ i \neq j \\ d(v_i, v_i) = 0 & \text{para } i = j, \ i = 1, \dots, n \end{cases}$$

**Definición.** Un punto  $x_0 \in N$  es una mediana absoluta de  $N$ , si para cada punto  $x \in N$  se tiene

$$\sum_{i=1}^n w_i d(v_i, x_0) \leq \sum_{i=1}^n w_i d(v_i, x) \quad (2.1)$$

En el problema de la mediana, los  $v_i \in V$  se corresponden a vértices de transporte (cruces de caminos, instalaciones,...) o a puntos de demanda que están conectados de forma única a la red del espacio real que se quiere modelar.

El siguiente teorema probado por Hakimi [10] es la principal caracterización de la localización de la mediana de una red y es una generalización del resultado de A.J. Goldstein, quien demostró que una mediana absoluta de un grafo en forma de árbol siempre se encontraba en un vértice.

**Teorema 2.1.** Una mediana absoluta de un grafo está siempre en un vértice del grafo.

*Demostración.* Para probar este teorema, se va a mostrar que si  $x_0$  es un punto arbitrario de  $N$  y  $x_0 \neq v_i$  para  $i = 1, 2, \dots, n$ , entonces siempre existe un vértice  $v_m$  en  $N$  tal que:

$$\sum_{i=1}^n w_i d(v_i, x_0) \geq \sum_{i=1}^n w_i d(v_i, v_m)$$

lo que probaría que un vértice de  $N$  será la mediana absoluta.

Sea  $x_0$  un punto del eje  $b(v_p, v_q)$  en  $N$ . Se sabe que:

$$d(v_i, x_0) = \min\{d(x_0, v_p) + d(v_p, v_i), d(x_0, v_q) + d(v_q, v_i)\}$$

Sea  $i_1, i_2, \dots, i_n$  una reorganización del conjunto de enteros  $1, 2, \dots, n$  tal que

$$\begin{cases} d(x_0, v_{i_k}) = d(x_0, v_p) + d(v_p, v_{i_k}) & \text{para } k = 1, 2, \dots, r, \ r \leq n \\ d(x_0, v_{i_k}) = d(x_0, v_q) + d(v_q, v_{i_k}) & \text{para } k = r+1, \dots, n \end{cases}$$

Por tanto, se puede escribir:

$$\sum_{i=1}^n w_i d(v_i, x_0) = \sum_{k=1}^r w_{i_k} [d(x_0, v_p) + d(v_p, v_{i_k})] + \sum_{k=r+1}^n w_{i_k} [d(x_0, v_q) + d(v_q, v_{i_k})] \quad (2.2)$$

A continuación, se va a estudiar la ecuación (2.2) diferenciando los dos siguientes posibles casos:

$$\text{Caso (a) : } \sum_{k=1}^r w_{i_k} \geq \sum_{k=r+1}^n w_{i_k}$$

$$\text{Caso (b) : } \sum_{k=1}^r w_{i_k} < \sum_{k=r+1}^n w_{i_k}$$



■ **Caso (a) :**

Se tiene que  $d(x_0, v_q) = d(v_p, v_q) - d(x_0, v_p)$ . Sustituyendo este valor en (2.2) se obtiene:

$$\sum_{i=1}^n w_i d(v_i, x_0) = \sum_{k=1}^r w_{i_k} [d(x_0, v_p) + d(v_p, v_{i_k})] + \sum_{k=r+1}^n w_{i_k} [d(v_p, v_q) - d(x_0, v_p) + d(v_q, v_{i_k})] \quad (2.3)$$

Por otro lado,  $d(v_q, v_p) + d(v_q, v_{i_k}) \geq d(v_p, v_{i_k})$ , y sustituyendo esta desigualdad en (2.3) se obtiene:

$$\begin{aligned} \sum_{i=1}^n w_i d(v_i, x_0) &\geq \sum_{k=1}^r w_{i_k} [d(x_0, v_p) + d(v_p, v_{i_k})] + \sum_{k=r+1}^n w_{i_k} [d(v_p, v_{i_k}) - d(x_0, v_p)] = \\ &= \sum_{i=1}^n w_i d(v_p, v_i) + \left[ \sum_{k=1}^r w_{i_k} - \sum_{k=r+1}^n w_{i_k} \right] d(x_0, v_p) \end{aligned}$$

De la hipótesis de este caso, se puede concluir que:

$$\sum_{i=1}^n w_i d(v_i, x_0) \geq \sum_{i=1}^n w_i d(v_p, v_i)$$

Y esto es lo que se quería demostrar.

■ **Caso (b) :**

El procedimiento para este caso es similar al caso anterior. Se sustituye en la ecuación (2.2) el valor  $d(x_0, v_p) = d(v_p, v_q) - d(v_q, x_0)$  y se obtiene:

$$\sum_{i=1}^n w_i d(v_i, x_0) = \sum_{k=1}^r w_{i_k} [d(v_p, v_q) - d(v_q, x_0) + d(v_p, v_{i_k})] + \sum_{k=r+1}^n w_{i_k} [d(x_0, v_q) + d(v_q, v_{i_k})] \quad (2.4)$$

Usando pasos similares al caso anterior, se reduce la ecuación (2.4) a la siguiente inecuación:

$$\sum_{i=1}^n w_i d(v_i, x_0) \geq \sum_{i=1}^n w_i d(v_i, v_q) + \left[ \sum_{k=r+1}^n w_{i_k} - \sum_{k=1}^r w_{i_k} \right] d(x_0, v_q)$$

Por último, haciendo uso de la hipótesis de este caso, se obtiene que:

$$\sum_{i=1}^n w_i d(v_i, x_0) > \sum_{i=1}^n w_i d(v_i, v_q)$$

que es lo que se quería demostrar.

□

Hasta ahora se ha considerado la localización de una sola instalación. A continuación, se va a suponer que  $p$  instalaciones tienen que ser posicionadas simultáneamente, donde  $p$  es un entero mayor que uno.

El problema de la  $p$ -mediana consiste en encontrar un conjunto de  $p$  puntos de  $N$  tal que la suma de las distancias ponderadas a los vértices sea mínima, es decir, se busca

$$\begin{aligned} \min \quad & F(S) = \sum_{i=1}^n w_i d(v_i, S) \\ \text{sujeto a} \quad & |S| = p \\ & S \subset N \end{aligned}$$

donde  $d(v_i, S) := \min\{d(v_i, s) \mid s \in S\}$  es la distancia entre  $v_i$  y  $S$ . En el problema de la  $p$ -mediana, cada vértice se asigna, por lo tanto, a la instalación abierta más cercana.

El principal resultado para el problema de la  $p$ -mediana lo obtuvo Hakimi en 1965 [11]. Demostró que el conjunto de lugares posibles para localizar las instalaciones puede restringirse al conjunto de vértices, es decir, la localización óptima de una instalación no se va a encontrar en un punto intermedio de un eje  $b(v_i, v_j) \in E$ . El argumento es el siguiente. En el punto óptimo, cada instalación debe estar localizada de forma que se minimice la suma de las distancias ponderadas a los vértices (clientes o puntos de demanda) que se le asignan. Por lo tanto, por el teorema 2.1, hay una configuración óptima de modo que cada instalación se encuentre en un vértice de la red.

## 2.4. El problema de la $p$ -mediana como problema de optimización entera

Dada la propiedad de optimalidad de los vértices, el problema de la  $p$ -mediana se puede expresar como un programa de optimización lineal entero mixto:

$$\min_{Y, X} F(Y, X) = \sum_{i=1}^n \sum_{j=1}^n w_i d_{ij} x_{ij} \quad (2.5a)$$

sujeto a

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (2.5b)$$

$$\sum_{j=1}^n y_j = p \quad (2.5c)$$

$$0 \leq x_{ij} \leq y_j, \quad i = 1, \dots, n, \quad j = 1, \dots, n \quad (2.5d)$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (2.5e)$$

donde  $d_{ij}$  es la distancia entre los vértices  $v_i$  y  $v_j$ . La variable  $y_j$  toma el valor 1 si  $v_j$  pertenece a la  $p$ -mediana, y 0 en caso contrario;  $Y$  denota el vector de las variables  $y_j$ . La variable  $x_{ij}$  es la fracción de demanda que surge en  $v_i$  satisfecha por la instalación  $v_j$  (si existe);  $X$  es la matriz formada por las variables  $x_{ij}$ .

Las restricciones (2.5b) significan que la demanda en  $v_i$  debe ser satisfecha por alguna instalación. En (2.5c) se asegura que exactamente  $p$  vértices pertenezcan a la  $p$ -mediana. Las restricciones (2.5d) evitan la asignación de demanda a un vértice que no esté en la  $p$ -mediana, además de garantizar la no negatividad. Por último, en (2.5e) se restringe a cada variable  $y_j$  a ser una variable binaria.

Nótese que el problema de la  $p$ -mediana tiene una solución óptima en la que todas las variables  $x_{ij}$  son enteras, es decir, tiene una solución óptima entera. Para cada vértice con demanda  $v_i$ , se puede asignar el valor  $x_{ij} = 1$  para un índice  $j$  tal que  $d_{ij} = \min\{d_{i\ell} \mid y_\ell = 1\}$ , y asignar el resto de variables  $x_{ik} = 0$ ,  $k \neq j$ . Esto corresponde a asignar la demanda de  $v_i$  al vértice más cercano  $v_j$  perteneciente a la  $p$ -mediana.

A continuación, se enuncia un resultado que muestra que resolver el problema de la  $p$ -mediana puede ser difícil en el peor de los casos.

**Teorema 2.2.** *El problema de la  $p$ -mediana es NP-hard.*

La demostración de este teorema se puede encontrar en [13].

## 2.5. Algoritmos

Aun siendo el problema de la  $p$ -mediana NP-hard, se han desarrollado muchos algoritmos eficientes y aproximaciones que en la práctica resuelven bien el problema de la  $p$ -mediana. Entre ellos, destacan los métodos heurísticos por dos razones. En primer lugar, son la única forma de abordar problemas

de gran dimensión. En segundo lugar, la mayoría de los procedimientos exactos convergen más rápidamente cuando hay buenas soluciones iniciales disponibles, que pueden ser proporcionadas por los algoritmos heurísticos.

Los principales métodos heurísticos se pueden dividir en dos grupos:

1. **Métodos heurísticos clásicos.** Pueden ser constructivos, como el método Greedy y el método Stingy, de búsqueda local, como el método del Vecindario y el método de Intercambio, o basados en programación matemática como el método de Lagrange.
2. **Métodos metaheurísticos.** Se van a presentar el método de búsqueda tabú, el método de búsqueda dispersa y el método híbrido.

A continuación se describen brevemente los métodos anteriores. Una descripción más detallada de los algoritmos puede verse en [16, 17, 19].

### 2.5.1. Método Greedy

El método Greedy fue propuesto por Kuehn y Hamburger en 1963 [15] y es el algoritmo más simple para resolver el problema de la  $p$ -mediana.

Sean

$$F(S) = \sum_{i=1}^n \min_{v_j \in S} d_{ij} \quad (2.6a)$$

$$\rho_j(S) \equiv F(S) - F(S \cup \{v_j\}), \text{ para } S \subseteq V \text{ y } v_j \in V \setminus S \quad (2.6b)$$

donde (2.6b) se corresponde con la disminución del valor de la función objetivo al añadir el vértice  $v_j$  al conjunto  $S$ , que representa el conjunto formado por las  $p$ -medianas.

El método comienza con un conjunto vacío  $S$  de instalaciones abiertas. A continuación, se resuelve el problema de la mediana en  $V$  y se agrega la instalación obtenida a  $S$ . Todas las instalaciones candidatas se examinan una por una, de manera que cada vez se selecciona aquella que reduce más la función objetivo y se añade al conjunto de medianas. El proceso continúa hasta que la solución incluye  $p$  instalaciones.

#### Pseudocódigo del algoritmo:

- **Paso 1:** Inicializar  $S$  al conjunto vacío;  $S := \emptyset$ .
- **Paso 2:** Seleccionar  $v_j$  tal que  $F(v_j) = \min_{v_k \in V} F(v_k)$ . Entonces  $S := \{v_j\}$ .
- **Paso 3:** Elegir  $v_j \in V \setminus S$  tal que  $\rho_j(S) = \max_{v_k \in V \setminus S} \rho_k(S)$ . Entonces  $S := S \cup \{v_j\}$ .
- **Paso 4:** Si  $|S| < p$ , volver al paso 3. En caso contrario, es decir, si  $|S| = p$ , el proceso termina.

No hay garantía de la optimalidad para este algoritmo, y su complejidad es del orden de  $O(p|V|^2)$ .

### 2.5.2. Método del Vecindario o Alterno

Maranzana propuso en 1964 [18] una mejora del método Greedy dando lugar al siguiente algoritmo.

El método comienza eligiendo una selección inicial de  $p$  vértices donde se localizarán las instalaciones abiertas, dando lugar al conjunto  $S^\circ$ . A continuación, cada vértice de  $V$  es asignado a la instalación más cercana de entre los  $p$  vértices de  $S^\circ$ , formando así  $p$  particiones de  $V$  a las que denominó vecindarios. Seguidamente, dentro de cada vecindario, se resuelve el problema de la mediana en  $V$ . Si al resolver cada problema no se cambia la localización de ninguna instalación, el proceso se detiene. En caso contrario, se reasignan los vértices de  $V$  a la instalación más cercana del nuevo conjunto de medianas formando los nuevos vecindarios y se repite el proceso. El algoritmo continúa hasta que no es posible mejorar más la solución.

#### Pseudocódigo del algoritmo

- **Paso 1:** Hacer  $t = 0$ , inicializar la solución  $S^t$  con cualquier conjunto de  $p$  vértices.
- **Paso 2:** Asignar cada vértice de  $V$  al punto más cercano de  $S^t$ , produciéndose una partición  $\{V_1^t, \dots, V_p^t\}$ .
- **Paso 3:** Para cada  $V_k^t$ ,  $k = 1, \dots, p$ , determinar la correspondiente mediana en  $V$ .
- **Paso 4:** Formar el conjunto  $S^{t+1}$  con las medianas de cada vecindario del paso 3. Se produce una nueva partición  $\{V_1^{t+1}, \dots, V_p^{t+1}\}$  de  $V$ .
- **Paso 5:** Si  $S^{t+1} \neq S^t$ , hacer  $t = t + 1$  y volver al paso 3. En caso contrario, ya se ha alcanzado una solución estable y el proceso termina.

Aunque el método es intuitivamente atractivo, el procedimiento tiene dos defectos principales. Por un lado, vértices lejanos incorporados en el conjunto inicial  $S^\circ$  a menudo pertenecen a los conjuntos intermedios subsiguientes  $S^t$ . Por otro, vértices que tienen un gran peso, una vez incluidos en  $S^t$  tienden a evitar la aparición de nuevas medianas. Por lo tanto, el rendimiento de este método depende en gran medida de la elección de  $S^\circ$ , lo que hace que sean necesarias varias ejecuciones con diferentes soluciones iniciales. Además, no se conoce ningún límite polinomial del tiempo de ejecución del método.

Esta heurística puede cambiar a un método exacto exhaustivo si todos los  $\binom{n}{p}$  subconjuntos posibles de vértices se eligen como una solución inicial. Sin embargo, este no suele ser el caso dado que la complejidad del algoritmo se incrementa en un  $O(n^p)$ .

### 2.5.3. Método Stingy

El método Stingy fue propuesto por Feldman, Lehrer y Ray en 1966 [7].

Se denota

$$\sigma_j(S) \equiv F(S \setminus \{v_j\}) - F(S), \text{ para cada } v_j \in S \subseteq V$$

al incremento en la función objetivo cuando se cierra la instalación localizada en  $v_j$ .

El método comienza con el conjunto de localizaciones de las instalaciones igual al conjunto de vértices. A continuación, se eliminan aquellas que incrementan menos la función objetivo, hasta obtener un conjunto de tamaño  $p$ .

#### Pseudocódigo del algoritmo:

- **Paso 1:** Inicializar el conjunto de localizaciones. Entonces  $S := V$  y  $F(S) = 0$ .
- **Paso 2:** Seleccionar  $v_j$  tal que  $\sigma_j(S) = \min_{v_k \in S} \sigma_k(S)$ . Entonces  $S := S \setminus \{v_j\}$ .
- **Paso 3:** Si  $|S| > p$  volver al paso 2. Si  $|S| = p$ , parar el algoritmo.

### 2.5.4. Método de intercambio

Teitz y Bart propusieron el método de intercambio en 1968 [24].

Se comienza con un patrón arbitrario de  $p$  instalaciones que definen un conjunto  $S$ . Se denota

$$\tau_{jk}(S) \equiv F(S) - F(S \setminus \{v_j\} \cup \{v_k\}) \text{ para } v_j \in S, v_k \in V \setminus S$$

a la variación en la función objetivo cuando la instalación localizada en  $v_j$  se mueve a  $v_k$  y el resto no cambian su posición.

El método consiste en eliminar un vértice que está en la solución y reemplazarlo por otro que no está en ella, en el caso de que este cambio mejore la función objetivo. El procedimiento termina cuando no existe tal intercambio que mejore la solución.

#### Pseudocódigo del algoritmo:

- **Paso 1:** Inicializar la solución  $S$  con un conjunto de  $p$  instalaciones.

- **Paso 2:** Seleccionar  $v_j \in S$  y  $v_k \in V \setminus S$ .
- **Paso 3:** Calcular  $\tau_{jk}(S)$ .  
 Si  $\tau_{jk}(S) > 0$ , entonces  $S := S \setminus \{v_j\} \cup \{v_k\}$  y volver al paso 2.  
 Si  $\tau_{jk}(S) \leq 0 \quad \forall j, k$ , entonces parar el algoritmo. Ningún cambio disminuye el valor de la función objetivo.

Claramente, el resultado del método depende de la configuración inicial de las instalaciones. Empíricamente, se obtienen buenos resultados empezando por la solución que se genera en el método Greedy o el método Stingy.

Por otro lado, cabe destacar que el número de iteraciones que se requieren depende del criterio elegido a la hora de seleccionar los pares de vértices  $\{v_j, v_k\}$ . En 1978, Nemhauser, Wolsey y Fisher [21] dieron un ejemplo en el cual, una elección pobre de los pares de vértices conllevaba una complejidad exponencial. En la práctica, una regla eficiente es seleccionar, para un vértice dado  $v_j$ , el vértice  $v_k$  tal que  $\tau_{jk}(S) = \max_{v_k \in V \setminus S} \tau_{j\ell}(S)$ .

### 2.5.5. Método de Lagrange

A continuación, se va a describir el algoritmo de relajación Lagrangiano para el problema de la  $p$ -mediana, el cual fue propuesto en 1977 por Cornuéjols, Fisher y Nemhauser [6] y por Narula, Ogbu y Samuelsson [20]. Este algoritmo tiene dos ventajas sobre los demás enfoques heurísticos:

1. En cada iteración se obtienen límites inferior y superior en el valor de la función objetivo.
2. Puede integrarse fácilmente en un algoritmo de ramificación y acotación para obtener posibles soluciones óptimas.

El método se basa en la relajación de la restricción (2.5b) del problema (2.5). Sea  $\lambda_i$  el multiplicador de Lagrange asociado a la restricción  $i$ -ésima. El problema relajado puede formularse como:

$$F(\Lambda) = \min_{Y, X} \sum_{i=1}^n \sum_{j=1}^n w_i d_{ij} x_{ij} + \sum_{i=1}^n \lambda_i \left( 1 - \sum_{j=1}^n x_{ij} \right)$$

sujeto a

$$\sum_{j=1}^n y_j = p$$

$$0 \leq x_{ij} \leq y_i, \quad i = 1, \dots, n, \quad j = 1, \dots, n$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n$$

donde  $\Lambda$  es el vector formado por los  $\lambda_i$ .

La función objetivo puede ser expresada como sigue:

$$F(\Lambda) = \sum_{i=1}^n \lambda_i - \max_{Y, X} \sum_{i=1}^n \sum_{j=1}^n (\lambda_i - w_i d_{ij}) x_{ij}$$

Las variables de asignación se determinan estableciendo:

$$x_{ij} = \begin{cases} y_j & \text{si } \lambda_i - w_i d_{ij} > 0 \\ 0 & \text{si } \lambda_i - w_i d_{ij} \leq 0 \end{cases}$$

Sea  $(\lambda_i - w_i d_{ij})^+ = \max\{\lambda_i - w_i d_{ij}, 0\}$ , entonces se define

$$\rho_j(\Lambda) = \sum_{i=1}^n (\lambda_i - w_i d_{ij})^+$$

Por lo tanto, el problema relajado se puede expresar de la siguiente forma:

$$F(\Lambda) = \sum_{i=1}^n \lambda_i - \max_Y \sum_{j=1}^n \rho_j(\Lambda) y_j$$

sujeto a

$$\sum_{j=1}^n y_j = p$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n$$

Para un vector dado  $\Lambda$ , las  $p$  instalaciones se localizan en los correspondientes vértices que tienen los  $p$  valores más grandes de  $\rho_j(\Lambda)$ , lo cual determina los valores de la variable de ubicación  $y_j$ . Como  $F(\Lambda)$  es una cota inferior del valor óptimo de (2.5) para cada  $\Lambda$ , la mejor cota posible viene dada por:

$$F_D = \max_{\Lambda} F(\Lambda)$$

que corresponde con el problema dual de Lagrange.

### 2.5.6. Método de búsqueda tabú ( $TS$ )

La búsqueda tabú es un algoritmo metaheurístico que aumenta el rendimiento del método de búsqueda local mediante el uso de estructuras de memoria. Consiste en marcar como tabú una solución potencial que ya ha sido determinada, de manera que el algoritmo no vuelva a pasar por ella. La lista tabú es la unidad de memoria a corto plazo en la que se guardan dichas soluciones, la cual suele tener una capacidad aleatoria.

Utiliza métodos de búsqueda local o de vecindario, es decir, se mueve iterativamente desde una solución  $S$  hasta otra  $S^*$  que se encuentra cerca. Denotando por  $N(S)$  al vecindario de cualquier solución  $S$  (un conjunto de instalaciones abiertas), el método consiste en ir moviéndose de una solución  $S$  hasta otra  $S^*$  que está en el vecindario  $N(S)$  sin volver a obtener las soluciones de la lista tabú.

### 2.5.7. Método de búsqueda dispersa ( $SS$ )

Este método fue propuesto por Glover, Laguna y Marty en el año 2000 [8]. Se basa en una estrategia en la cual se tiene un conjunto moderado de buenas soluciones, al cual se denomina conjunto de referencia, que evoluciona principalmente combinando sus soluciones para construir otras nuevas.

En el algoritmo diseñado para resolver el problema de la  $p$ -mediana, se introdujo una distancia en el espacio de soluciones para controlar la diversificación del método. Se define la distancia entre dos soluciones  $S_1$  y  $S_2$  como:

$$\eta(S_1, S_2) = \sum_{v_i \in S_1} \min_{v_j \in S_2} d_{ij} + \sum_{v_j \in S_2} \min_{v_i \in S_1} d_{ij}$$

El conjunto de referencia está formado por las  $k$  mejores soluciones (donde  $k$  es un parámetro) y las  $r - k$  soluciones elegidas al azar siguiendo algunos criterios de diversificación. Por lo tanto,  $r$  denota el tamaño del conjunto de referencia. Las soluciones de un subconjunto seleccionado del conjunto de referencia se combinan de la siguiente manera:

- En primer lugar, se encuentra el conjunto de instalaciones que aparecen en cada solución del subconjunto.
- A continuación, para obtener un conjunto de tamaño  $p$ , se agregan nuevas instalaciones de forma iterativa según las normas predefinidas.
- Las soluciones combinadas son mejoradas mediante la búsqueda local basada en intercambios.
- Las soluciones resultantes se incorporan al conjunto de referencia debido a que mejoran una de las  $k$  mejores soluciones o debido a que mejoran la diversidad del conjunto de acuerdo con la distancia entre sus soluciones.

### 2.5.8. Método híbrido

Este método combina ideas de varias metaheurísticas puras. Se trata de un enfoque de inicio múltiple donde cada iteración consiste en la construcción de puntos iniciales mediante el método Greedy aleatorio seguido de una búsqueda local.

Como en los métodos  $TS$  y  $SS$ , se encuentra una ruta entre dos soluciones para un conjunto de buenas soluciones o soluciones de élite y se realiza una búsqueda local a partir de cada solución en esa ruta.

Comparado con otros métodos, su procedimiento a menudo proporciona mejores resultados, tanto en términos de tiempo de ejecución, como de calidad de la solución.

## 2.6. Variantes del problema de la $p$ -mediana

### 2.6.1. El problema de la $p$ -mediana en forma de árbol

Como ya se ha mencionado anteriormente, el problema de la  $p$ -mediana puede ser resuelto en un tiempo polinomial cuando está planteado en forma de árbol. Para ver este hecho, primero se introduce el concepto de árbol.

**Definición.** Una red se dice que tiene forma de árbol cuando cualquier par de vértices distintos de la red están unidos mediante un único camino simple.

Goldman propuso en 1971 [9] un algoritmo polinomial para obtener la mediana en un árbol. Está claro que si cualquier vértice del árbol tiene la mitad o más de la mitad de la demanda total, es óptimo localizar una instalación en él. En caso de no hacerlo, la instalación se alejaría de la mitad o más de la demanda y esto aumentaría el valor de la función objetivo. Por lo tanto, se supone que ninguno de los vértices del árbol se encuentra en esta situación.

Sea  $J = \{1, \dots, j, \dots, m\}$  el conjunto de las  $m$  posibles localizaciones de las instalaciones y sea  $I = \{1, \dots, i, \dots, n\}$  el conjunto de los  $n$  clientes o puntos de demanda. Sea  $w_i$  la demanda del cliente  $i$  y  $c_{ij}$  el coste por satisfacer al cliente  $i$  desde la instalación  $j$ .

Denominando vértice terminal a cualquier vértice que está conectado a un único vértice del árbol, el algoritmo analiza cada vértice terminal y aquellos a los que están unidos. Para ello, se define  $w'_i$  como la demanda modificada del vértice  $i \in I$  cuando se elimina el vértice terminal que estaba conectado con él; es decir, la demanda modificada de un vértice es la suma de la demanda del propio vértice y la del vértice terminal que se elimina. Por otro lado, se denota

$$D_{total} = \sum_{i \in I} w_i$$

que es la demanda total del árbol.

#### Pseudocódigo:

- **Paso 1:** Inicializar  $w'_i = w_i \forall i \in I$ .
- **Paso 2:** Seleccionar  $i \in I$  tal que  $i$  es vértice terminal y sea  $k \in I$  el vértice unido a él.
- **Paso 3:** Eliminar  $i$  y el eje  $(i, k)$ . Entonces  $w'_k = w'_k + w'_i$ .
- **Paso 4:** Si  $w'_k \geq \frac{D_{total}}{2}$ , parar el algoritmo,  $k$  es una mediana. En caso contrario, volver al paso 2.

Se está ante un algoritmo de orden  $O(n)$ , ya que el paso 2 se puede realizar en un tiempo constante y cada vértice se examina como máximo una vez.

En 1979, Kariv y Hakimi presentaron un algoritmo para el problema de la  $p$ -mediana en forma de árbol con un orden de  $O(n^2 p^2)$ . Posteriormente, en 1996, Tamir [23] mejoró el tiempo de cálculo y presentó un algoritmo para el mismo problema con un orden de  $O(pn^2)$ .

### 2.6.2. Planificación de instalaciones con capacidad

El objetivo de este problema sigue siendo colocar  $p$  instalaciones de forma que se minimice la distancia media ponderada entre los vértices de demanda y los lugares en los que se colocan las instalaciones, pero en este caso, se tienen restricciones por las que las instalaciones pueden atender a lo sumo una demanda preestablecida.

Sean  $w_i$ ,  $d_{ij}$  e  $y_j$ , como en el apartado 2.4. Sea  $C$  la demanda máxima que una instalación puede atender, es decir, su capacidad. Definimos la variable decisión  $z_{ij}$  como la cantidad de demanda en el vértice  $v_i$  que es atendida por la instalación situada en  $v_j$ . El problema puede formularse como:

$$\min_{Y,X} F(Y,X) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} z_{ij} \quad (2.7a)$$

sujeto a

$$\sum_{j=1}^n z_{ij} = w_i, \quad i = 1, \dots, n \quad (2.7b)$$

$$\sum_{j=1}^n y_j = p \quad (2.7c)$$

$$z_{ij} \geq 0, \quad i = 1, \dots, n, j = 1, \dots, n \quad (2.7d)$$

$$z_{ij} \leq w_i y_j, \quad i = 1, \dots, n, j = 1, \dots, n \quad (2.7e)$$

$$\sum_{i=1}^n z_{ij} \leq C, \quad j = 1, \dots, n \quad (2.7f)$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (2.7g)$$

La función objetivo (2.7a) indica que se quiere minimizar la distancia ponderada por la demanda sumada sobre todas las instalaciones y clientes. Las restricciones (2.7b) implican que un vértice debe satisfacer toda su demanda, y (2.7c) que se deben colocar  $p$  instalaciones. En las restricciones (2.7d) se indica que la demanda atendida debe ser no negativa, y en (2.7e) que un cliente puede ser atendido en una instalación si está abierta. Por último, (2.7f) restringen la demanda que puede atender cada instalación y (2.7g) fuerzan a  $y_j$  a ser variables binarias.

### 2.6.3. El problema de localización de instalaciones con costes fijos

En este modelo se supone que abrir la instalación  $j$  conlleva unos costes fijos  $f_j$  y que asignar el cliente  $i$  a la instalación  $j$  conlleva unos costes variables de producción y transporte  $c_{ij}$ .

El problema consiste en encontrar el número de instalaciones y sus localizaciones, así como la asignación de los clientes a ellas, para minimizar el coste total. Con las variables  $x_{ij}$  e  $y_j$  anteriormente definidas, se puede formular el problema como el siguiente programa lineal entero mixto:

$$\min_{Y,X} F(Y,X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n f_j y_j \quad (2.8a)$$

sujeto a

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (2.8b)$$

$$0 \leq x_{ij} \leq y_j, \quad i = 1, \dots, n, j = 1, \dots, n \quad (2.8c)$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (2.8d)$$

donde las restricciones (2.8b) significan que la demanda en  $v_i$  debe ser satisfecha por alguna instalación, las restricciones (2.8c) evitan la asignación de demanda a un vértice que no esté en la  $p$ -mediana y además se aseguran de garantizar la no negatividad. Por último, en (2.8d) se restringe a cada variable  $y_j$  a ser binaria.



El problema puede ser resuelto con las mismas técnicas que el problema de la  $p$ -mediana, y al igual que él, es NP-hard en un grafo general y puede ser resuelto en tiempo polinomial usando el algoritmo de Kolen's [14] cuando está planteado en forma de árbol.

#### 2.6.4. Problema de cobertura máxima

En algunos casos, se puede estar interesado en que cada vértice  $v_i$ , para  $i = 1, \dots, n$ , esté dentro de una distancia máxima  $\bar{d}_i$  para poder garantizar un nivel mínimo de servicio de calidad. Este hecho lleva a reemplazar la restricción (2.5b) por  $\sum_{v_j \in V_i} x_{ij} = 1$  donde  $V_i := \{v_j \in V \mid d_{ij} \leq \bar{d}_i\}$ . Tal y como señalaron Feldman, Lehrer y Ray en 1966 [7], el problema mantiene la estructura de la  $p$ -mediana siempre que  $d_{ij}$  sea reemplazado por un número arbitrariamente grande cuando se tenga que  $v_j \notin V_i$ . Por otro lado, probaron que la propiedad de optimalidad de los vértices ya no era válida para el problema. Sin embargo, en 1977, Church y Meadows [4] demostraron que el conjunto de vértices complementado por los puntos situados en los ejes a distancia  $\bar{d}_i$  desde el vértice  $v_i$  para  $i = 1, \dots, n$ , siempre contenía una configuración óptima para localizar las instalaciones. Por último, cabe destacar que si se obtiene un valor infinito para la función objetivo significa que no hay solución al problema con una distancia máxima  $\bar{d}_i$  y  $p$  instalaciones.

En 1974, Church y Reville [3] sugirieron usar un sustituto de la función objetivo para maximizar la cantidad de clientes atendidos, dando lugar al problema de máxima cobertura. En esencia, se está minimizando el número de demandas que no se llegan a cubrir, lo que equivale a maximizar el número de demandas cubiertas. Primero se define el parámetro  $w_{ij}$  como sigue:

$$w_{ij} = \begin{cases} w_i & \text{si } d_{ij} \leq \bar{d}_i \\ 0 & \text{si } d_{ij} > \bar{d}_i \end{cases}$$

Entonces, el planteamiento del problema como programa lineal entero queda:

$$\max_{Y, X} F(Y, X) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_{ij} \quad (2.9a)$$

sujeto a

$$\sum_{j=1}^n x_{ij} \leq 1, \quad i = 1, \dots, n \quad (2.9b)$$

$$\sum_{j=1}^n y_j = p \quad (2.9c)$$

$$0 \leq x_{ij} \leq y_j, \quad i = 1, \dots, n, \quad j = 1, \dots, n \quad (2.9d)$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (2.9e)$$

Es decir, las restricciones son las mismas que en el problema de la  $p$ -mediana, salvo (2.9b), que indica que ya no es necesario que se suministre a todos los clientes.

Este problema puede ser reducido a un problema de la  $p$ -mediana si se considera una instalación ficticia  $n+1$  con  $w_{i,n+1} = 0$  para  $i = 1, \dots, n$ . La instalación ficticia estaría cerca de todos los puntos de demanda, de forma que cuando  $\sum_{j=1}^n x_{ij} < 1$ , necesariamente se tenga  $y_{n+1} = 1$ . Sin embargo, esto no significa que la demanda que no se satisfacía antes se satisfaga ahora, ya que la instalación es ficticia y se sigue disponiendo solo de  $p$  instalaciones. Para poder convertirlo en un problema de la  $p$ -mediana se definen los siguientes parámetros:

$$\hat{d}_{ij} = \begin{cases} 1 & \text{si } d_{ij} \leq \bar{d}_i \\ 0 & \text{si } d_{ij} > \bar{d}_i \end{cases}$$

$$\hat{d}_{i,n+1} = 0, \quad i = 1, \dots, n$$

$$\hat{w}_i = -w_i, \quad i = 1, \dots, n$$

El problema se plantea entonces como:

$$\min_{Y, X} F(Y, X) = \sum_{i=1}^n \sum_{j=1}^{n+1} \hat{w}_i \hat{d}_{ij} x_{ij} \quad (2.10a)$$

sujeto a

$$\sum_{j=1}^{n+1} x_{ij} = 1, \quad i = 1, \dots, n \quad (2.10b)$$

$$\sum_{j=1}^{n+1} y_j = p + 1 \quad (2.10c)$$

$$0 \leq x_{ij} \leq y_j, \quad i = 1, \dots, n, \quad j = 1, \dots, n+1 \quad (2.10d)$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (2.10e)$$

$$y_{n+1} = 1 \quad (2.10f)$$

de manera que las restricciones (2.10c) y (2.10f) aseguran que se localizan las  $p$  instalaciones disponibles, y la  $p+1$  siempre es la ficticia.

## Capítulo 3

# Aplicaciones

### 3.1. Introducción

El objetivo de este capítulo es presentar la aplicación del problema de la  $p$ -mediana a la posible ubicación de ciertas instalaciones en Aragón, así como presentar un trabajo reciente, publicado en una revista de investigación, sobre la instalación de estaciones de recarga de vehículos eléctricos.

### 3.2. Una aplicación en la Comunidad Autónoma de Aragón

En esta sección, se va a resolver el problema de la  $p$ -mediana usando *CPLEX* para el caso particular de localizar instalaciones en la Comunidad Autónoma de Aragón. Para ello, se necesita conocer los clientes, las localizaciones candidatas para las instalaciones, la demanda de los clientes, las distancias entre las posibles localizaciones de las instalaciones y el número de instalaciones que se desea instalar. Se ha considerado la localización de  $p$  instalaciones que podrán ubicarse en las capitales de las 33 comarcas de la comunidad. La demanda asociada a cada capital de comarca se define como el número total de habitantes de la comarca, es decir, la suma de los habitantes de todos los pueblos de la comarca. En la figura 3.1 se puede ver la distribución de las 33 comarcas de la comunidad.

#### 3.2.1. Recopilación de datos

En primer lugar, se han obtenido los datos sobre el número de habitantes en cada comarca de la comunidad. Para ello, se ha buscado en la página web del Instituto Aragonés de Estadística, dentro del padrón municipal de habitantes, en la cual se encuentran las cifras oficiales de población según el lugar de residencia. Además, dicha información se puede extraer en una tabla Excel. La fecha más reciente disponible es el año 2017 y esa es la que se ha extraído. La dirección de la página web es: <http://servicios3.aragon.es/iaeaxi/tabla.do?path=/02/01/01/01/&file=0201010101.px&type=pcaxis&L=0>.

A continuación, se ha obtenido cuál es la capital de cada una de las comarcas. Cabe destacar, que en algunos casos, la comarca dispone de capital administrativa y cultural, y que se ha tomado siempre la capital administrativa.

Una vez recopilada la información anterior, la cual puede verse en el anexo, se ha calculado la distancia que separa a cada una de las capitales de comarca, para así poder construir la matriz de distancias. Para ello, se ha utilizado ‘*VRP Solver*’. *VRP Solver* es un complemento de Excel desarrollado por Günes Erdogan, que puede descargarse libremente en la página web <http://people.bath.ac.uk/ge277/index.php/vrp-spreadsheet-solver/>. Este complemento permite representar, resolver y visualizar los resultados de los problemas de planificación de ruta de vehículos (VRP). Además, permite calcular matrices de distancias de hasta 200 localidades usando direcciones y códigos postales. Para el cálculo de las distancias, *VRP Solver* utiliza ‘*Bing Maps*’, y por lo tanto, hay que tener conexión a internet en todo momento. Además, se necesita una clave de *Bing Maps*, la cual se puede generar libremente



Figura 3.1: Comunidad Autónoma de Aragón

en la siguiente dirección: <https://www.bingmapsportal.com/>.

Para poder aplicar este complemento, el primer paso ha sido obtener la información sobre los códigos postales de cada capital de comarca. Una vez obtenidos, se introducen en la hoja Excel correspondiente. El programa calcula primero la longitud y la latitud de cada capital. Una vez que se tienen esos datos, ya se puede calcular la distancia (kilómetros) que separa a cada una de ellas. La apariencia del archivo Excel puede verse en las imágenes 3.2 y 3.3. Cuando se tienen todas las distancias calculadas, se extrae la información en forma de documento de texto, para luego poder utilizarlo en *CPLEX*.

### 3.2.2. Creación del modelo

Como se ha mencionado anteriormente, para resolver el problema se ha utilizado *CPLEX*, un programa que permite resolver de forma eficaz problemas expresados como modelos de programación matemática. En este caso, se ha utilizado la versión de *CPLEX* 12.6.2.

El modelo desarrollado, ‘*pmedian.mod*’, puede verse en la figura 3.4. Además, se ha creado el fichero ‘*pmedian.dat*’ que contiene la información relevante sobre el problema, es decir, el vector de clientes, el vector de localizaciones, la matriz de distancias, el vector de demandas y el valor de  $p$ .

	A	B	C	D	E
1	Location ID	Name	Address	Latitude (y)	Longitude (x)
2	0	01 La Jacetania	22700, Jaca	42,5710945	-0,5447870
3	1	02 Alto Gállego	22600, Sabiñanigo	42,5237808	-0,3681730
4	2	03 Sobrarbe	22330, Ainsa	42,4145088	0,1396600
5	3	04 La Ribagorza	22430, Graus	42,1903763	0,3363350
6	4	05 Cinco Villas	50600, Ejea de los Caballeros	42,0831146	-1,1091070
7	5	06 Hoya de Huesca / Plana de Uesca	22001, Huesca	42,1392632	-0,4040100
8	6	07 Somontano de Barbastro	22300, Barbastro	42,0211487	0,1088970
9	7	08 Cinca Medio	22400, Monzón	41,8940811	0,2213180
10	8	09 La Litera / La Llitera	22500, Binefar	41,8463402	0,2880300
11	9	10 Los Monegros	22200, Sarinena	41,7747383	-0,1545580
12	10	11 Bajo Cinca / Baix Cinca	22520, Fraga	41,4894257	0,2105700
13	11	12 Tarazona y el Moncayo	50500, Tarazona	41,8927498	-1,6899480
14	12	13 Campo de Borja	50540, Borja	41,8338089	-1,5311290
15	13	14 Aranda	50250, Illueca	41,5321388	-1,6340700
16	14	15 Ribera Alta del Ebro	50630, Alagon	41,7742844	-1,1215700
17	15	16 Valdejalón	50100, La Almunia de doña Godina	41,4751167	-1,3389510
18	16	17 D.C. Zaragoza	50001, Zaragoza	41,6504173	-0,8777700
19	17	18 Ribera Baja del Ebro	50770, Quinto	41,4282799	-0,5019560
20	18	19 Bajo Aragón-Caspe / Baix Aragó-Caspe	50700, Caspe	41,2763557	-0,0175690
21	19	20 Comunidad de Calatayud	50300, Calatayud	41,3513641	-1,6507061
22	20	21 Campo de Cariñena	50400, Cariñena	41,3573189	-1,2066950
23	21	22 Campo de Belchite	50130, Belchite	41,3236847	-0,7127350
24	22	23 Bajo Martín	44520, Híjar	41,1407500	-0,4048250

Figura 3.2: Latitudes y longitudes de las capitales de comarca

	A	B	C	D
1	From	To	Distance	Duration
2	01 La Jacetania	01 La Jacetania	0,00	0:00
3	01 La Jacetania	02 Alto Gállego	17,40	0:16
4	01 La Jacetania	03 Sobrarbe	70,88	0:57
5	01 La Jacetania	04 La Ribagorza	121,47	1:43
6	01 La Jacetania	05 Cinco Villas	125,57	1:58
7	01 La Jacetania	06 Hoya de Huesca / Plana de Uesca	71,92	0:52
8	01 La Jacetania	07 Somontano de Barbastro	122,90	1:23
9	01 La Jacetania	08 Cinca Medio	137,69	1:29
10	01 La Jacetania	09 La Litera / La Llitera	143,98	1:32
11	01 La Jacetania	10 Los Monegros	125,77	1:44
12	01 La Jacetania	11 Bajo Cinca / Baix Cinca	218,67	2:28
13	01 La Jacetania	12 Tarazona y el Moncayo	226,39	2:34
14	01 La Jacetania	13 Campo de Borja	201,62	2:05
15	01 La Jacetania	14 Aranda	227,25	2:23
16	01 La Jacetania	15 Ribera Alta del Ebro	164,23	1:43
17	01 La Jacetania	16 Valdejalón	194,53	2:08
18	01 La Jacetania	17 D.C. Zaragoza	142,39	1:35
19	01 La Jacetania	18 Ribera Baja del Ebro	185,47	1:56
20	01 La Jacetania	19 Bajo Aragón-Caspe / Baix Aragó-Caspe	236,62	2:46
21	01 La Jacetania	20 Comunidad de Calatayud	224,11	2:18
22	01 La Jacetania	21 Campo de Cariñena	193,41	1:58
23	01 La Jacetania	22 Campo de Belchite	193,09	2:00
24	01 La Jacetania	23 Bajo Martín	222,76	2:33

Figura 3.3: Distancia entre las capitales de comarca

```

//Datos
int P = ...;
{string} Clientes = ...;
{string} Almacenes = ...;
int Demanda[Clientes] = ...;
float Distancia[Clientes][Almacenes] = ...;

//Variables
dvar boolean AbroAlmacen[Almacenes];
dvar boolean AsignoCliente[Clientes][Almacenes];

//Objetivo
minimize
    sum( c in Clientes , w in Almacenes )
        Demanda[c]*Distancia[c][w]*AsignoCliente[c][w];

//Restricciones
subject to {
    forall( c in Clientes )
        sum( w in Almacenes )
            AsignoCliente[c][w] == 1;

    sum( w in Almacenes )
        AbroAlmacen[w] == P;

    forall( c in Clientes , w in Almacenes )
        AsignoCliente[c][w] <= AbroAlmacen[w];
}

```

Figura 3.4: Modelo de la  $p$ -mediana en *CPLEX*

### 3.2.3. Evaluación de los resultados

Se han llevado a cabo varias ejecuciones del modelo para diferentes valores de  $p$  con objeto de poder evaluar y comparar los resultados obtenidos. Para ello, se va cambiando el valor de  $p$  en el fichero ‘*pmedian.dat*’ antes de cada ejecución del modelo. Se ha utilizado un ordenador personal Intel (R) Core (TM) i3-4030U a 1.90 GHz con 4.00 GB de RAM.

- Para  $p = 3$ : Como era de esperar, las tres instalaciones se localizan en las capitales de cada provincia, es decir, en Zaragoza, Huesca y Teruel, ya que son las ciudades con mayor demanda. La asignación de los clientes a las respectivas instalaciones abiertas es la siguiente:
  - Huesca: La Jacetania, Alto Gállego, Sobrarbe, La Ribagorza, Hoya de Huesca, Somontano de Barbastro, Cinca Medio, La Litera y Los Monegros.
  - Teruel: Cuencas Mineras, Jiloca, Comunidad de Teruel, Maestrazgo, Sierra de Albarracín y Gúdar-Javalambre.
  - Zaragoza: Cinco Villas, Bajo Cinca, Tarazona y el Moncayo, Campo de Borja, Aranda, Ribera Alta del Ebro, Valdejalón, D.C. Zaragoza, Ribera Baja del Ebro, Bajo Aragón-Caspe, Comunidad de Calatayud, Campo de Cariñena, Campo de Belchite, Bajo Martín, Campo de Daroca, Andorra-Sierra de Arcos, Bajo Aragón y Matarraña.

Cabe destacar, que cada comarca es asignada a su capital de provincia, salvo el Bajo Cinca, que está en Huesca y es asignada a la instalación de Zaragoza. El tiempo total de ejecución del modelo es de 0,09 segundos y el valor de la función objetivo es 33.396.587,51.

- Para  $p = 4$ : La instalación que se añade se localiza en Alcañiz. Esto cambia la asignación de las comarcas, que pasa a ser la siguiente:

- Alcañiz: Bajo Cinca, Bajo Aragón-Caspe, Bajo Martín, Andorra-Sierra de Arcos, Bajo Aragón, Maestrazgo y Matarraña.
- Huesca: La Jacetania, Alto Gállego, La Ribagorza, Sobrarbe, Hoya de Huesca, Somontano de Barbastro, Cinca Medio, La Litera, y Los Monegros.
- Teruel: Jiloca, Cuencas Mineras, Comunidad de Teruel, Sierra de Albarracín y Gúdar-Javalambre.
- Zaragoza: Cinco Villas, Tarazona y el Moncayo, Campo de Borja, Aranda, Ribera Alta del Ebro, Valdejalón, D.C. Zaragoza, Ribera Baja del Ebro, Comunidad de Calatayud, Campo de Cariñena, Campo de Belchite y Campo de Daroca.

Con respecto al caso anterior, se observa que el Maestrazgo pasa de ser asignado a Teruel a ser asignado a Alcañiz, y además, se reparten las comarcas asignadas anteriormente a Zaragoza entre ella y Alcañiz. El tiempo total de ejecución es de 0,05 segundos y el valor de la función objetivo disminuye hasta un total de 26.952.745,83.

- Para  $p = 6$ : Se añaden las nuevas instalaciones en La Almunia de Doña Godina y en Monzón. La reasignación de las comarcas es la siguiente:
  - Alcañiz: Bajo Aragón-Caspe, Bajo Martín, Andorra-Sierra de Arcos, Bajo Aragón, Maestrazgo y Matarraña.
  - Huesca: La Jacetania, Alto Gállego y Hoya de Huesca.
  - La Almunia de Doña Godina: Tarazona y el Moncayo, Campo de Borja, Aranda, Valdejalón, Comunidad de Calatayud, Campo de Cariñena y Campo de Daroca.
  - Monzón: Sobrarbe, La Ribagorza, Somontano de Barbastro, Cinca Medio, La Litera, Los Monegros y Bajo Cinca.
  - Teruel: Jiloca, Cuencas Mineras, Comunidad de Teruel, Sierra de Albarracín y Gúdar-Javalambre.
  - Zaragoza: Cinco Villas, Ribera Alta del Ebro, D.C. Zaragoza, Ribera Baja del Ebro y Campo de Belchite.

Se observa que las comunidades que antes estaban asignadas a Zaragoza se han repartido entre esta y La Almunia de Doña Godina. Lo mismo pasa con Monzón y Huesca. El tiempo de ejecución del modelo es de 0,06 segundos y la función objetivo se reduce hasta un valor de 17.897.550,65, es decir, es casi la mitad del valor que tomaba la función objetivo para  $p = 3$ .

- Para  $p = 10$ : Las instalaciones se localizan en Teruel, Alcañiz, Calatayud, Zaragoza, Borja, Fraga, Monzón, Huesca, Ejea de los Caballeros y Sabiñánigo. El reparto es el mismo para Alcañiz, sin embargo, el resto cambian:
  - Borja: Tarazona y el Moncayo y Campo de Borja.
  - Calatayud: Aranda, Valdejalón, Comunidad de Calatayud, Campo de Daroca y Jiloca.
  - Ejea de los Caballeros: Cinco Villas.
  - Fraga: Bajo Cinca.
  - Huesca: Hoya de Huesca.
  - Monzón: La Ribagorza, Somontano de Barbastro, Cinca Medio, La Litera y Los Monegros.
  - Sabiñánigo: La Jacetania, Alto Gállego y Sobrarbe.
  - Teruel: Cuencas Mineras, Comunidad de Teruel, Sierra de Albarracín y Gúdar-Javalambre.
  - Zaragoza: Ribera Alta del Ebro, D.C. Zaragoza, Ribera Baja del Ebro, Campo de Cariñena

y Campo de Belchite.

Cabe destacar, que en este caso no se ha seleccionado La Almunia de Doña Godina para localizar una instalación. El tiempo total de ejecución es 0,05 segundos y el valor de la función objetivo es 10.318.652,56.

- Para  $p = 20$ : Las instalaciones se localizan en Teruel, Alcañiz, Andorra, Utrillas, Calamocha, Calatayud, Caspe, Quinto, Zaragoza, La Almunia de Doña Godina, Alagón, Borja, Fraga, Sariñena, Monzón, Barbastro, Huesca, Ejea de los Caballeros, Graus y Jaca. La distribución de las comarcas es la misma que en el caso anterior para Fraga, Huesca, Borja y Ejea de los Caballeros. Para el resto es la siguiente:

- Alcañiz: Bajo Aragón y Matarraña.
- Andorra: Bajo Martín, Andorra-Sierra de Arcos y Maestrazgo.
- Alagón: Ribera Alta del Ebro.
- Barbastro: Somontano de Barbastro.
- Calamocha: Campo de Daroca y Jiloca.
- Calatayud: Aranda y Comunidad de Calatayud.
- Caspe: Bajo Aragón-Caspe.
- Graus: Sobrarbe y La Ribagorza.
- Jaca: La Jacetania y Alto Gállego.
- La Almunia de Doña Godina: Valdejalón y Campo de Cariñena.
- Monzón: Cinca Medio y La Litera.
- Quinto: Ribera Baja del Ebro y Campo de Belchite.
- Sariñena: Los Monegros.
- Teruel: Comunidad de Teruel, Sierra de Albarracín y Gúdar-Javalambre.
- Utrillas: Cuencas Mineras.
- Zaragoza: D.C. Zaragoza.

Cabe destacar, que Sabiñánigo no se selecciona en este caso. Además, se observa que prácticamente todas las comarcas se asignan a sus respectivas capitales. El tiempo necesario de resolución es de 0,06 segundos y el valor de la función objetivo es 3.211.895,8.

- Para  $p = 33$ : Al tener 33 clientes, obtenemos un valor de la función objetivo igual a cero, así como la localización de las instalaciones en cada una de las capitales y la asignación de cada comarca a su respectiva capital. El tiempo de ejecución es de 0,01 segundos.

### 3.3. Una aplicación a la localización de estaciones de recarga

En 2016, se publicó en [12] un estudio sobre la planificación de las localizaciones de estaciones de carga para vehículos eléctricos públicos (EV) en Beijing, China. En él se compararon las localizaciones óptimas de tres modelos de ubicación diferentes; el modelo de cubrimiento de conjuntos (SCP), el modelo de máximo cubrimiento (MCLP) y el modelo de la  $p$ -mediana (PMP). En cada modelo se trató de incorporar las limitaciones locales de la oferta y la demanda de las estaciones de carga. Hay que destacar que en esta aplicación los factores sociodemográficos y el acceso a las respectivas localizaciones tenía un impacto directo en el consumo de EV y, por lo tanto, se tuvo en cuenta a la hora de determinar las localizaciones.



### 3.3.1. Demanda y localizaciones

Para estimar la demanda, se realizaron entrevistas con las partes interesadas y se utilizó el método Delphi de clasificación con datos agregados de los anuarios estadísticos municipales y el censo nacional. De esta forma, se concluyó que en las zonas interiores de la ciudad estaban más interesados en la compra de coches eléctricos, lo cual generaba una mayor demanda de estaciones de carga. En total, se consideraron 313 puntos de demanda.

Para estimar las posibles localizaciones, se analizaron las restricciones institucionales y espaciales que se podían tener a la hora de construir las infraestructuras. De este modo, se incluían como posibles ubicaciones las estaciones de repostaje de gasolina de SINOPEC y CNPC, y los aparcamientos públicos de dos grandes compañías de estacionamiento, dando un total de 1029 posibles localizaciones.

Una vez conocidos dichos datos, se utilizaron los tres modelos nombrados anteriormente para obtener las localizaciones óptimas.

### 3.3.2. Resultados

Para resolver los tres modelos, se utilizó CPLEX 12.6.1. como solucionador del problema de programación lineal entero, y todas las pruebas computacionales se realizaron en un ordenador personal Intel (R) Core (TM) i5-4570 a 3.20 GHz con 16.0 GB de RAM.

En primer lugar, se estableció para los dos primeros modelos la distancia de cobertura crítica en 5 km. Sin embargo, para esta distancia, el modelo SCP no podía atender algunos puntos de demanda, y por tanto, se eliminaron del modelo las restricciones asociadas a esos puntos de demanda para poder garantizar la factibilidad del problema. A continuación, se resolvió el modelo SCP para los puntos de demanda restantes y se obtuvo que eran suficientes 118 estaciones de carga para cubrir los puntos de demanda. Para los otros dos modelos, se tomó el valor de  $p$  igual a 118 y se obtuvieron las localizaciones óptimas que minimizaban sus respectivas funciones objetivo.

Se observó que SCP y MCLP producían resultados muy similares, en los que 83 de 118 de sus localizaciones eran las mismas. Ambos modelos sugerían estaciones más dispersas. Por otro lado, el PMP tenía resultados muy diferentes, teniendo solo 26 y 22 estaciones en común con SCP y MCLP respectivamente. El PMP sugería estaciones cercanas a puntos con alta demanda. Se observó que las estaciones de carga del PMP cubrían de forma significativa más puntos de demanda dentro de un rango de distancia más corto, entre 1- 4 km, en comparación con las soluciones del SCP y MCLP. Sin embargo, para una distancia de cobertura crítica de 5 km o más, las estaciones de SCP y MCLP cubrían más puntos que las de PMP, pero sus demandas totales cubiertas eran muy parecidas. Esto indicaba que las soluciones de los tres modelos podían cubrir niveles similares de demanda, dentro de la distancia de cobertura de 5 km.

En este estudio, la accesibilidad de las estaciones de carga era una de las grandes preocupaciones por el hecho de que si la distancia a la estación más cercana era grande, los usuarios potenciales de EV podían abandonar la adopción de los mismos. Por este motivo, desde esta perspectiva, se concluyó que el PMP parecía más eficaz para determinar la localización de las estaciones de carga de los EV.



# Bibliografía

- [1] M.S. BAZARAA Y J.J. JARVIS Y H.D. SHERALI, *Linear Programming and Network Flows*, Wiley-Interscience, Hoboken, NJ, 3rd edition, 2005.
- [2] M.S. BAZARAA Y H.D. SHERALI Y C.M. SHETTY, *Nonlinear Programming. Theory and Algorithms*, Wiley-Interscience, Hoboken, NJ, 3rd edition, 2006.
- [3] R.L. CHURCH Y C.S. REVELLE (1974), The maximal covering location problem, *Papers in Regional Science Association*, Volume 32, 101-118.
- [4] R.L. CHURCH Y M.E. MEADOWS (1979), Location modelling utilizing maximum service distance criteria, *Geographical Analysis*, Volume 11, 358-373.
- [5] M. CONFORTI Y G. CORNUÉJOLS Y G. ZAMBELLI, *Integer Programming*, Graduate Texts in Mathematics, Springer, New York, 2014.
- [6] G. CORNUÉJOLS Y M.L. FISHER Y G.L. NEMHAUSER (1977) , Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms, *Management Science*, Volume 23, 789-810.
- [7] E. FELDMAN Y E.A. LEHRER Y T.L. RAY (1966), Warehouse locations under continuous economies of scale, *Management Science*, Volume 12, 670-684.
- [8] F. GLOVER Y M. LAGUNA Y R. MARTY (2000), Fundamentals of scatter search and path relinking, *Control and Cybernetics*, 39 (3), 653-684.
- [9] A.J. GOLDMAN (1971), Optimal center location in simple networks, *Transportation Science*, Volume 5, 212-221.
- [10] S.L. HAKIMI (1964), Optimum locations of switching centers and the absolute centers and medians of a graph, *Operations Research*, Volume 12, 450-459.
- [11] S.L. HAKIMI (1965), Optimum distribution of switching centers in a communication network and some related graph theoretic problems, *Operations Research*, Volume 13, 462-475.
- [12] S.Y. HE Y Y.H. KUO Y D. WU, Incorporating institutional and spatial factors in the selection of the optimal location of public electric vehicle charging facilities: A case study of Beijing, China, *Transportation Research*, Part C 67 (2016), 131-148.
- [13] O. KARIV Y S.L. HAKIMI (1979), An algorithmic approach to network location problems, Part II: The  $p$ -medians, *SIAM Journal on Applied Mathematics*, Volume 37, 539-560.
- [14] A. KOLEN (1983), Solving covering problems and the uncapacitated plant location problem on trees, *European Journal of Operational Research*, Volume 12, 266-278.
- [15] A.A KUEHN Y M.J. HAMBURGER (1963), A heuristic program for locating warehouses, *Management Science*, Volume 9, 643-666.

- [16] M. LABBÉ Y D. PEETERS Y J.F. THISSE, *Chapter 7: Location on Networks*, M.O. BALL Y T.L. MAGNANTI Y C.L. MONMA Y G.L. NEMHAUSER, *Handbooks in Operations Research and Management Science*, (1995), Volume 8, 551-624.
- [17] G. LAPORTE Y S. NICKEL Y F. SALDANHA DA GAMA. *Location Science*, Springer, 2015.
- [18] E.E MARANZANA (1964), On the location of supply points to minimize transport costs, *Operational Research Quarterly*, Volume 15, 261-270.
- [19] N. MLADENOVIC Y J. BRIMBERG Y P. HANSEN Y J.A. MORENO-PÉREZ (2007), The  $p$ -median problem: A survey of metaheuristic approaches, *European Journal of Operational Research*, Volume 179, 927-939.
- [20] S.C. NARULA Y U.I. OGBU Y H.M. SAMUELSSON (1977), An algorithm for the  $p$ -median problem, *Operations Research*, Volume 25, 709-713.
- [21] G.L. NEMHAUSER Y L.A. WOLSEY Y M.L. FISHER (1978), An analysis of approximations for maximizing submodular set functions-I, *Mathematical Programming*, Volume 14, 265-294.
- [22] G.L. NEMHAUSER Y L.A. WOLSEY. *Integer and Combinatorial Optimization*. Wiley, New York, 1999.
- [23] A. TAMIR (1996), An  $O(pn^2)$  algorithm for the  $p$ -median and related problems on tree graphs, *Operations Research Letters*, Volume 19, 59-64.
- [24] M.B. TEITZ Y E. BART (1968), Heuristic methods for estimating the generalized vertex median of a weighted graph, *Operations Research*, Volume 16, 955-961.

# Anexo

En la figura 3.5 se pueden ver los datos recogidos para poder resolver el problema de la  $p$ -mediana para el caso de la Comunidad Autónoma de Aragón.

	A	B	C	D
1	COMARCAS	POBLACIÓN	CAPITAL	CÓDIGO POSTAL
2	01 La Jacetania	17.744	Jaca	22700
3	02 Alto Gállego	13.480	Sabiñánigo	22600
4	03 Sobrarbe	7.317	Aínsa	22330
5	04 La Ribagorza	12.067	Graus	22430
6	05 Cinco Villas	30.557	Ejea de los Caballeros	50600
7	06 Hoya de Huesca / Plana de Uesca	67.393	Huesca	22001
8	07 Somontano de Barbastro	23.652	Barbastro	22300
9	08 Cinca Medio	23.760	Monzón	22400
10	09 La Litera / La Llitera	18.209	Binéfar	22500
11	10 Los Monegros	18.794	Sariñena	22200
12	11 Bajo Cinca / Baix Cinca	24.470	Fraga	22520
13	12 Tarazona y el Moncayo	13.842	Tarazona	50500
14	13 Campo de Borja	13.965	Borja	50540
15	14 Aranda	6.828	Illueca	50250
16	15 Ribera Alta del Ebro	27.016	Alagón	50630
17	16 Valdejalón	28.340	La Almunia de Doña Godina	50100
18	17 D.C. Zaragoza	744.579	Zaragoza	50001
19	18 Ribera Baja del Ebro	8.634	Quinto	50770
20	19 Bajo Aragón-Caspe / Baix Aragó-Casp	14.459	Caspe	50700
21	20 Comunidad de Calatayud	37.700	Calatayud	50300
22	21 Campo de Cariñena	10.062	Cariñena	50400
23	22 Campo de Belchite	4.660	Belchite	50130
24	23 Bajo Martín	6.447	Hijar	44530
25	24 Campo de Daroca	5.660	Daroca	50360
26	25 Jiloca	12.490	Calamocha	44200
27	26 Cuencas Mineras	8.230	Utrillas	44760
28	27 Andorra-Sierra de Arcos	10.339	Andorra	44500
29	28 Bajo Aragón	28.708	Alcañiz	44600
30	29 Comunidad de Teruel	45.937	Teruel	44001
31	30 Maestrazgo	3.209	Cantavieja	44140
32	31 Sierra de Albarracín	4.483	Albarracín	44126
33	32 Gúdar-Javalambre	7.460	Mora de Rubielos	44400
34	33 Matarraña / Matarranya	8.259	Valderrobres	44580

Figura 3.5: Comarcas, capitales, códigos postales y población